

VIRTUAL LABYRINTH

Pflichtenheft



Inhaltsverzeichnis

Pflichtenheft.....	1
Inhaltsverzeichnis.....	2
1 Zielbestimmungen.....	3
1.1 Muss-Kriterien.....	3
2 Produkteinsatz.....	4
2.1 Zielgruppe	4
2.2 Betriebsbedienungen.....	4
3 Produktumgebung.....	5
3.1 Software.....	5
3.2 Hardware.....	5
3.3 Orgware.....	5
4 Produktfunktionen.....	6
4.1 Start Playing.....	6
4.3 Generate Labyrinth.....	8
4.4 Save Highscore.....	9
4.6 Time Runs Out.....	9
4.7 Use Special Ability.....	10
4.8 Display Snapshot.....	11
4.9 Singleplayer-Modus.....	11
4.10 Hauptmenü.....	11
4.11 Serverseitige Aufgaben.....	11
4.12 Klassendiagramm.....	12
5 Produktdaten.....	12
6 Produktleistungen.....	13
6.1 Labyrinth generieren.....	13
6.2 Spielen.....	13
6.3 Punktestand.....	13
7 Benutzeroberfläche.....	14
7.1 Mainmenu.....	14
7.2 Singleplayer.....	15
7.3 Spieloberfläche.....	16
8 Qualitätszielbestimmungen.....	18
9 Globale Testszenarien und Testfälle.....	19
10 Entwicklungsumgebung.....	19
11 Ergänzungen.....	20

1 Zielbestimmungen

1.1 Muss-Kriterien

- Das Spiel
 - Ziel des Spieles ist es, einen Weg aus dem Labyrinth zu finden.
 - Das Labyrinth ist ein 2-dimensionales Spielfeld, welches aus Wegen und Mauern besteht. Auf Ersteren können Gegenstände liegen.
 - Am Beginn des Spieles muss der Spieler folgende Einstellungen vornehmen:
 - Welchen Schwierigkeitsgrad soll das Spiel haben?
 - Mit welchem Charakter soll gespielt werden?
 - In welche Richtung soll das Labyrinth generiert werden? (Nord, Süd, Ost, West)
 - Es stehen 3 Charaktere zur Verfügung:
 - „Nuck Chorris“
 - „Master of Zen“
 - „The Old Man“
 - Der Schwierigkeitsgrad wirkt sich auf folgende Parameter aus:
 - Die Anzahl der Gegenstände die im Labyrinth verteilt liegen (je höher der Schwierigkeitsgrad ist, desto weniger Gegenstände können im Labyrinth gefunden werden)
 - Die Anzahl an Verzweigungen beziehungsweise Sackgassen im Labyrinth (je schwieriger das Spiel ist desto mehr Verzweigungen / Sackgassen gibt es im Labyrinth)
 - Gegenstände werden zufällig im Labyrinth verteilt und können nur jeweils einmal benutzt werden.
 - Es gibt folgende Gegenstände:
 - „Bomb“: ist eine Bombe, die bei Benützung eine Mauer im Labyrinth sprengt
 - „Activator“: ermöglicht es dem Spieler die Fähigkeit seines Helden einzusetzen (sie aktiviert also quasi die Spezialfähigkeit des Helden)
 - „Squirrel“: es gibt 3 Arten von „Squirrels“ (es wird zufällig bestimmt welches „Squirrel“ beim Laufen über den „Squirrel-Gegenstand“ ausgelöst wird)
 - „Good“: ein dem Spieler gut gesinntes „Squirrel“ läuft voraus und zeigt dem Spieler richtigen Weg. Es wird jedoch immer schneller und verschwindet wenn es außer Sichtweite gelangt.
 - „Evil“: ein „Evil Squirrel“ läuft in eine Sackgasse wo es anschließend verschwindet. Es sieht gleich aus wie ein Good Squirrel.
 - „Crazy“: ein „Crazy Squirrel“ läuft auf den Spieler zu und versucht ihn zu fressen. Erreicht es den Spieler wird dieser gefressen und das Spiel endet.
 - „Random Box“: eine „Random Box“ beinhaltet einen zufälligen Gegenstand („Bomb“, „Squirrel“ oder „Activator“) der beim Darüberlaufen erscheint und aktiviert wird (im Falle einer Bombe

wird diese natürlich nicht aktiviert, sondern der Spieler verfügt nun über eine Bombe)

- Die Helden haben folgenden Fähigkeiten
 - „Nuck Chorris“ kann mit seinem „Hound-Rouse-Kick“ eine Mauer zerstören
 - Die Spezialfähigkeit von „The Old Man“ bewirkt, dass der richtige Weg für einen, von der Spielfeldgröße abhängigen, Zeitraum (z.B.: 10 Sekunden) per Pfeil am Bildschirm angezeigt wird.
 - Die Spezialfähigkeit von „Master of Zen“ erhöht den Sichtbereich im Labyrinth für kurze Zeit (abhängig vom Schwierigkeitsgrad).
- Die Effekte der Helden können nur aktiviert werden wenn zuvor ein „Activator“ im Labyrinth gefunden wurde.
- Die Spielmodi können im Hauptmenü ausgewählt werden. Es existieren folgende Spielmodi:
 - „Arcade“: Der Spieler muss vor Ablauf der Zeitfrist am Ziel ankommen.
 - „Heroin Squirrel“: Am Beginn des Spieles erscheint vor dem Spieler ein „Squirrel“, das vor dem Spieler davonläuft. Ziel dieses Modus ist es das „Squirrel“ zu fangen bevor es aus dem Labyrinth entwischt (je höher der Schwierigkeitsgrad, desto schneller läuft das „Squirrel“).
- Um zu verhindern dass ein Spieler ohne einen entsprechenden Gegenstand benutzt zu haben durch eine Mauer läuft, werden sogenannte Checkpoints generiert.
- Ein Checkpoint ist ein Kreuzungspunkt an dem es mehr als eine Weg-Alternative gibt. Erreicht der Spieler einen nicht gültigen Checkpoint ist das Spiel beendet.
- Ein Checkpoint gilt dann als „nicht gültig“, wenn er kein benachbarter des zuletzt erreichten ist. Dadurch wird gewährleistet dass durch keine Mauer gegangen wurde, ergo auch nicht geschummelt wurde (Wenn der Spieler durch eine Mauer läuft und somit den Weg durch das Labyrinth abkürzt, erreicht er ja zwangsläufig einen nicht gültigen Checkpoint und das Spiel wird somit beendet).
- Um den Spieler langfristig zu motivieren werden die erreichten Punkte in einer der Datenbank auf einem Server eingetragen.

2 Produkteinsatz

2.1 Zielgruppe

Unsere zu erreichende Zielgruppe sind all jene Leute, die sich gerne bewegen und dies mit dem natürlich vorhandenen Spieltrieb verbinden wollen.

2.2 Betriebsbedingungen

- Das Spiel soll 24/7 verfügbar und
- wartungsfrei sein, sowie
- über eine automatisch verwaltete „Highscore-Liste“ verfügen.

3 Produktumgebung

Folgende Produkte beziehungsweise Gegenstände müssen vorhanden sein, um unser Produkt nutzen zu können:

3.1 Software

- Client
 - Windows Mobile
 - Android
- Server
 - IIE Server
 - Apache Server

3.2 Hardware

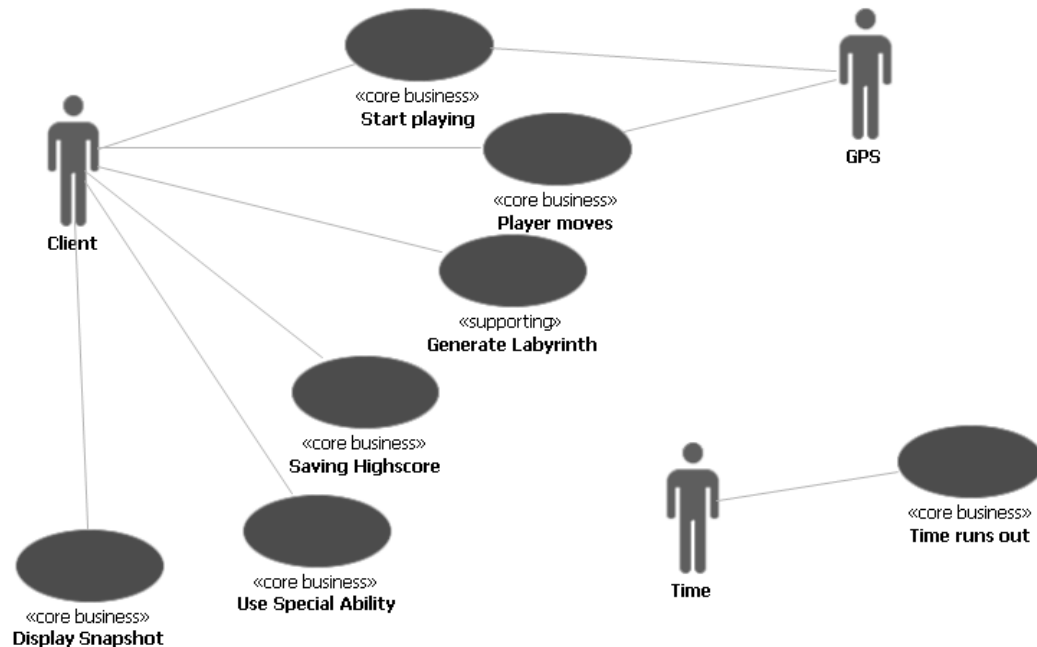
- Client
 - GPS-Fähiges Mobiles Endgerät mit Internetanbindung
- Server
 - Internet fähiger Server

3.3 Orgware

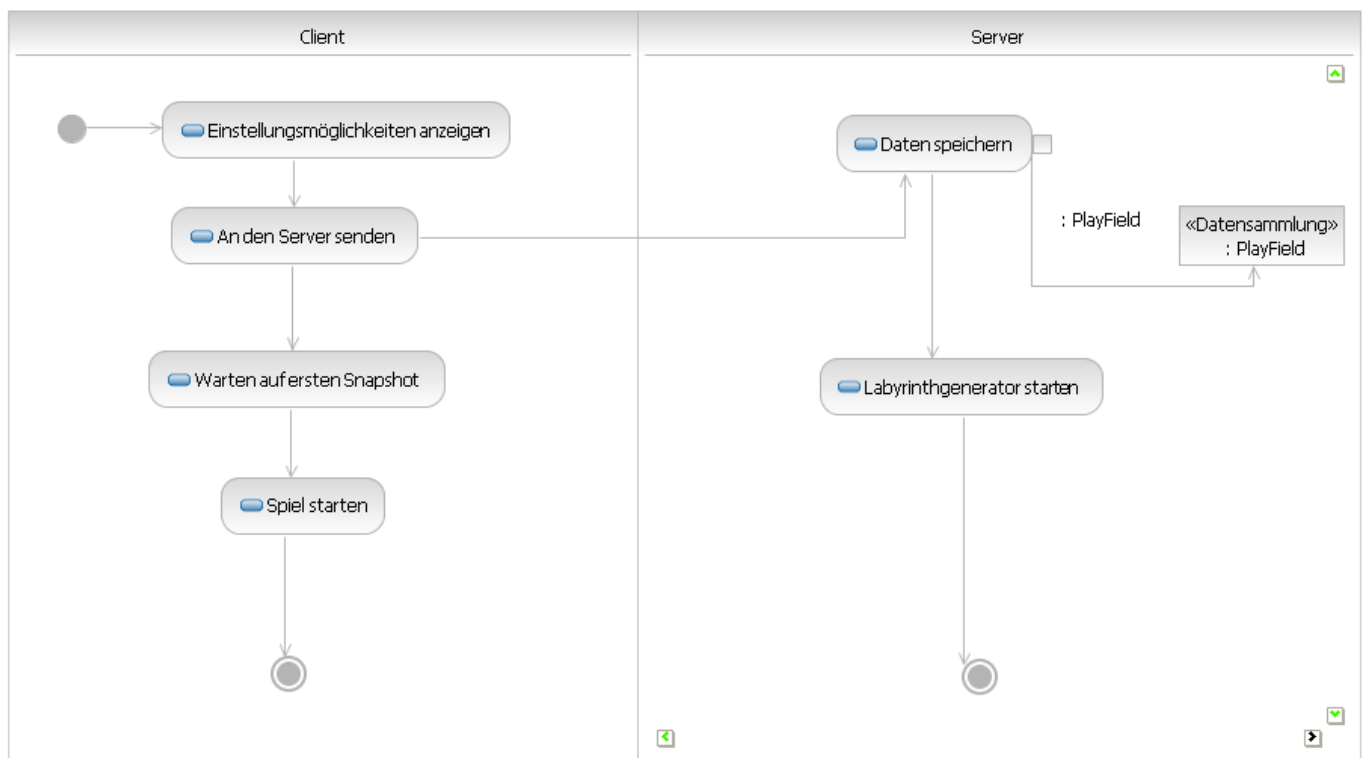
- Gewährleistung der permanenten Internetanbindung

4 Produktfunktionen

Übersicht über den Aufbau:



4.1 Start Playing



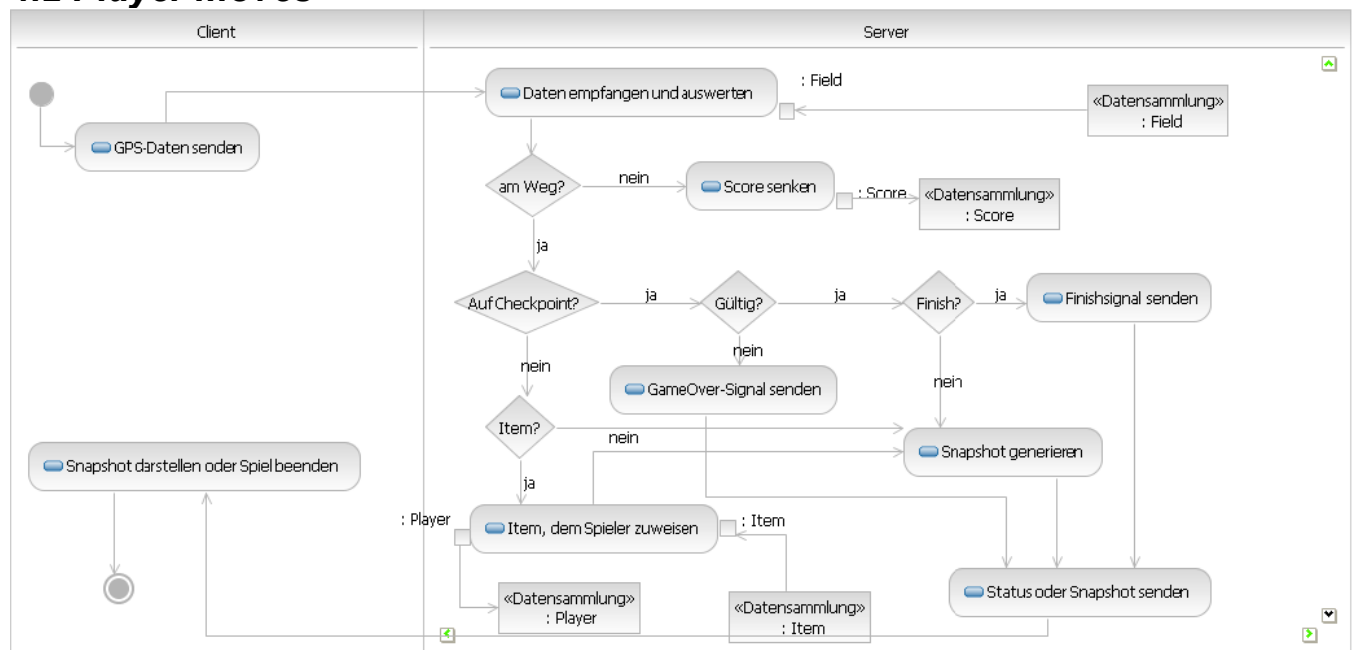
Die Einstellungsmöglichkeiten werden am Bildschirm Client angezeigt (Bild der Oberfläche siehe weiter unten – Punkt 7.2) und die getroffenen Spezifikationen

werden an den Server gesendet. Die Daten werden gespeichert und der Labyrinthgenerator wird gestartet.

Sobald das Labyrinth generiert ist und also in einem ersten Snapshot gespeichert wurde, sendet der Server dem Client den ersten darzustellenden Snapshot und das Spiel ist somit startbereit.

- Einstellungsmöglichkeiten am Beginn:
 - Größe des Spielfelds in Metern (Durchmesser)
 - Durchmesser
 - oder: Höhe, Breite
 - Schwierigkeitsgrad
 - Leicht
 - Mittel
 - Schwer
 - Spielcharakter
 - Position im Spielfeld
 - Eckpunkte
 - Mitte

4.2 Player moves



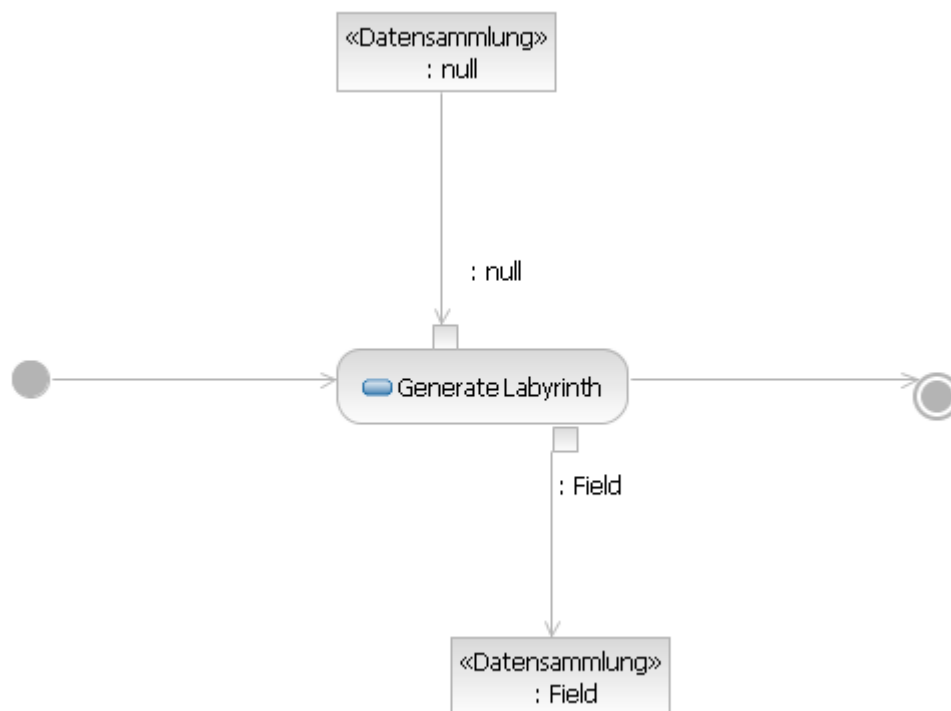
Das empfangene GPS-Signal wird an den Server gesendet, der die Position nun auswertet und einen entsprechenden Snapshot generiert, der dann dem Client zurückgesendet wird. Sobald der Client den Snapshot erhalten hat, wird er dargestellt, oder alternativ, falls kein Snapshot empfangen wurde (sondern das Finish Signal) wird das Spiel beendet.

Beim Generieren des Snapshots müssen folgende Fälle überprüft werden:

- 1) Der Spieler befindet sich am Weg.
 - Die Position des Spielers wird im Snapshot aktualisiert.
- 2) Der Spieler erreicht den aktuellen Checkpoint.

- Position des Spielers wird im Snapshot aktualisiert.
 - Der nächste Checkpoint wird aktiviert. (Spezialfall: Bei mehreren möglichen Wegen müssen alle eventuell „nächsten“ Checkpoints aktiviert werden)
 - Der zuvor erreichte Checkpoint wird deaktiviert.
- 3) Der Spieler erreicht den letzten Checkpoint im Labyrinth (Ziel).
- Finish-Status wird an den Client gesendet.
- 4) Der Spieler befindet sich außerhalb des Wegs.
- Der Spieler wird aufgefordert sich wieder auf den Weg zu begeben, da sonst das Spielende (in Form eines „Game-Over“) droht.
 - Der Highscore wird gesenkt, solange der Spieler sich in der Mauer befindet.
- 5) Der Spieler erreicht einen deaktivierten Checkpoint.
- Spiel wird beendet. (durch „Game-Over“).
- 6) Der Spieler erreicht einen aktiven Checkpoint auf dem sich ein Gegenstand befindet.
- Der Gegenstand wird dem Spieler zugeordnet, aus der Datenbank entfernt und das Gegenstandsereignis ausgelöst.

4.3 Generate Labyrinth



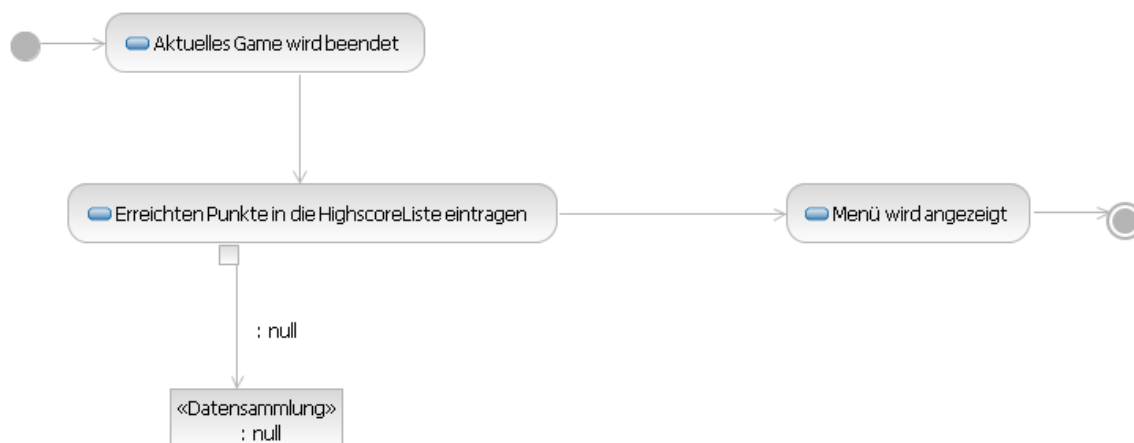
Mithilfe der Datensammlung, in der sich die abgespeicherten Spezifikationen aus 4.1 befinden, soll hier ein auf Zufall basierendes Labyrinth generiert werden. Dieses Labyrinth soll dann in der Datensammlung „Field“ gespeichert werden.

4.4 Save Highscore



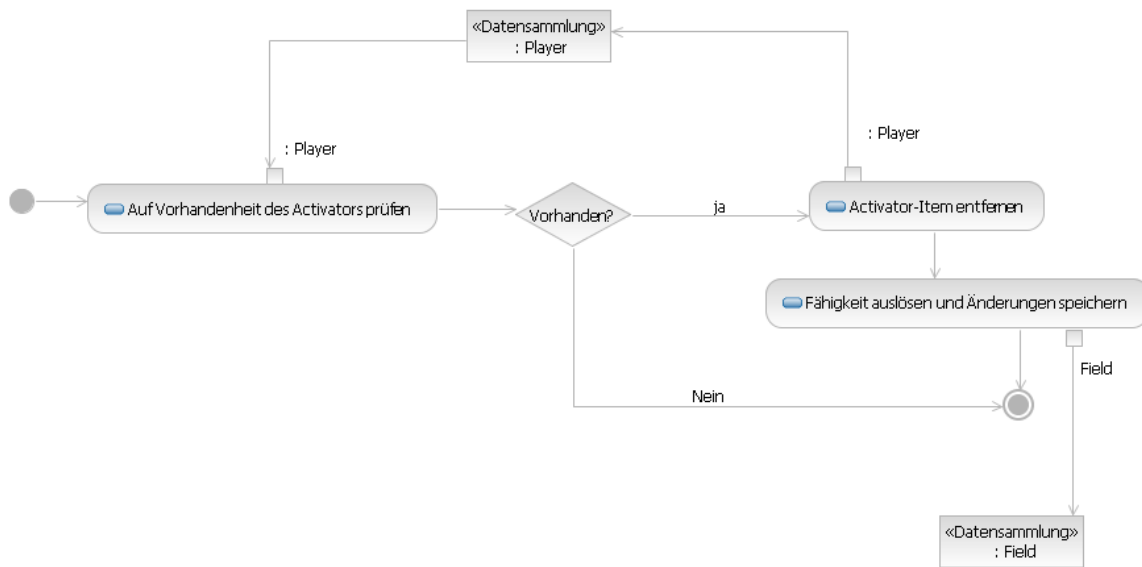
Der Punktestand eines Spieles wird nach Beenden des Spiels aus einer Datensammlung am Client ausgelesen (der Highscore wird in sie beim auslaufen der Zeit, bzw. beim Game-Over oder Win eingetragen). Der erreichte Score wird anschließend an den Server gesendet, der ihn in seiner Highscoreliste aktualisiert.

4.6 Time Runs Out



Dieser Use Case wird ausgeführt sobald die Zeit, welche der Spieler zum Bewältigen des Labyrinths hat, abläuft. Das Spiel wird beendet und ein Menü (Auswahlmöglichkeiten: „erneut Versuchen“, „Zurück zum Hauptmenü“) angezeigt. Die erreichten Punkte werden wie in 4.4 in einer Datensammlung am Client gespeichert.

4.7 Use Special Ability



Die Spezialfähigkeit ist abhängig vom Charakter und kann nur benutzt werden, wenn zuvor ein „Activator-Gegenstand“ gefunden wurde.

Hier wird also beim Benützen eines Items überprüft, ob ein „Activator“ vorhanden ist (dies erfährt man aus der Datensammlung „Player“, die alle Items und spielerrelevanten Informationen (wie z.B. dessen Namen) beinhaltet).

Ist ein „Activator“ vorhanden, wird er entfernt, die Fähigkeiten des Charakters ausgelöst und die daraus resultierenden Änderungen auf das Spielfeld werden in einem „Field“ gespeichert.

Ist kein „Activator“ vorhanden, geschieht nichts.

Es existieren folgende „Special-Abilities“:

„Nuck Chorris“:

- „Houseroundkick“ (Nach Drücken der Aktivierungstaste wird die nächste Mauer in die der Spieler läuft zu einem Weg)

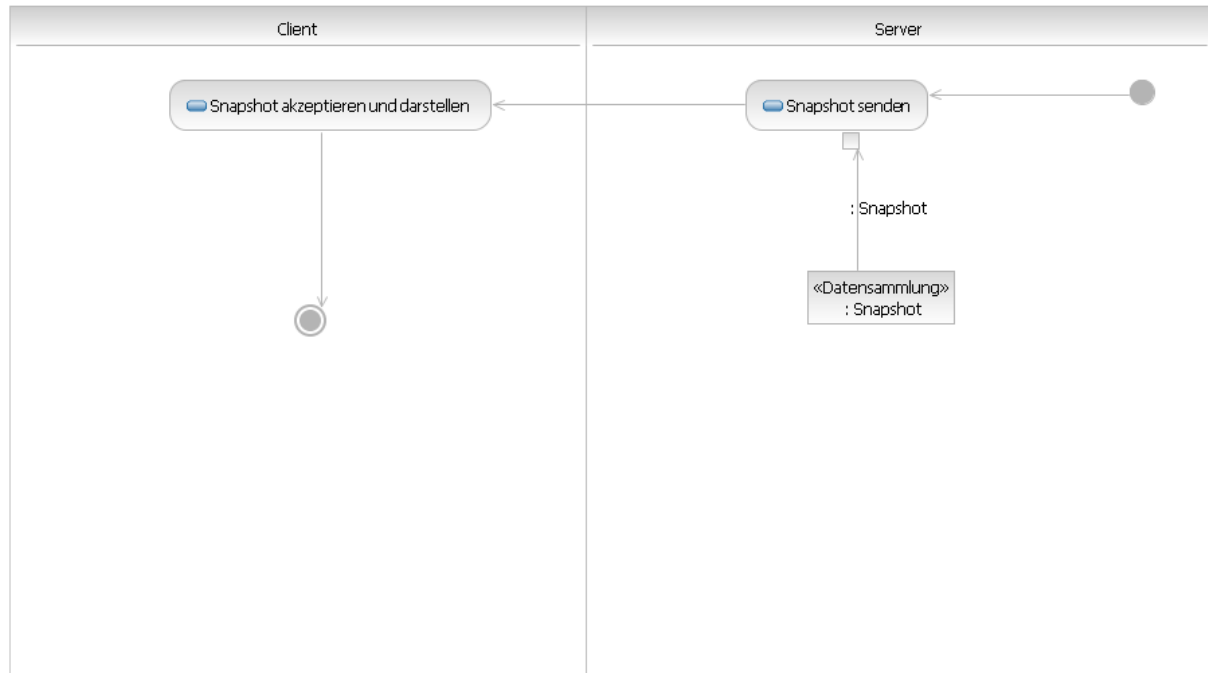
„Crazy Magician“:

- „Eye of Truth“ (Erweitert das Blickfeld)

„The Old Man“:

- „Master of Zen“ (Der richtige Weg wird für einen, von der Spielfeldgröße abhängigen, Zeitraum (z.B.:10 Sekunden) per Pfeil am Bildschirm angezeigt)

4.8 Display Snapshot



Ein „Snapshot“ (entspricht im Wesentlichen einem am Server generierten Abbild des Spielfelds) wird am Server generiert und an den Client gesendet, der ihn dann am Bildschirm darstellt.

4.9 Singleplayer-Modus

Der Benutzer spielt das Produkt im „Singleplayer-Modus“ - das heißt, dass er alleine, ohne weitere Mitspieler, spielt.

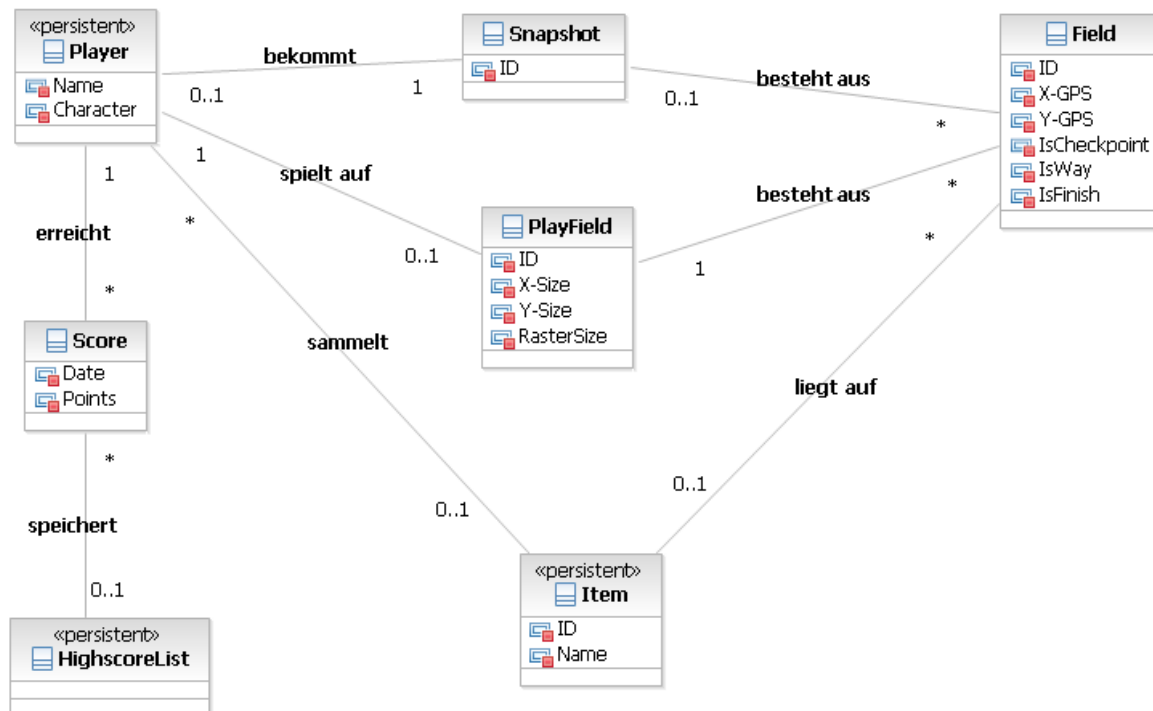
4.10 Hauptmenü

Im Hauptmenü soll der Spieler zwischen „Singleplayer“, „Multiplayer“, „Options“ und „Quit“ wählen können.
(weitere Informationen: Siehe 7.1 Hauptmenü)

4.11 Serverseitige Aufgaben

Am Server wird der erreichte „Highscore“ gespeichert.

4.12 Klassendiagramm



5 Produktdaten

Unser Produkt befindet sich auf dem mobilen Endgerät und ermöglicht es so dem Spieler, durch Spielen unseres Spiels, absolut einzigartige Erfahrungen im Wandeln durch die Umwelt, kombiniert mit dem Lösen eines kniffligen Labyrinths zu sammeln. Das Labyrinth wird am Client generiert und in der Datensammlung „Playfield“ gespeichert.

Für nähere Infos siehe Punkte 4.0 – 4.9.

6 Produktleistungen

6.1 Labyrinth generieren

Unser Produkt generiert am Client selbstständig, durch den Usecase „Generate Labyrinth“, das Labyrinth (siehe 4.3).

6.2 Spielen

Die GPS Koordinaten des Spielers werden laufend aktualisiert und am Bildschirm visualisiert.

Im Spiel unterscheidet unser Produkt im Wesentlichen 5 Fälle:

1. Spieler befindet sich auf dem Weg → nichts unternehmen
2. Spieler befindet sich in einer Wand → Highscore wird reduziert
3. Spieler befindet sich auf einem gültigen Checkpoint → nichts unternehmen
4. Spieler befindet sich auf einem ungültigen Checkpoint → Spiel beenden
5. Spieler erreicht den Ausgang → Spiel beenden und den Highscore speichern

6.3 Punktestand

Im Spiel erreicht man einen gewissen Punktestand der in einer „Highscorelist“ gespeichert wird. Diese befindet sich am Server.

7 Benutzeroberfläche

7.1 Mainmenu



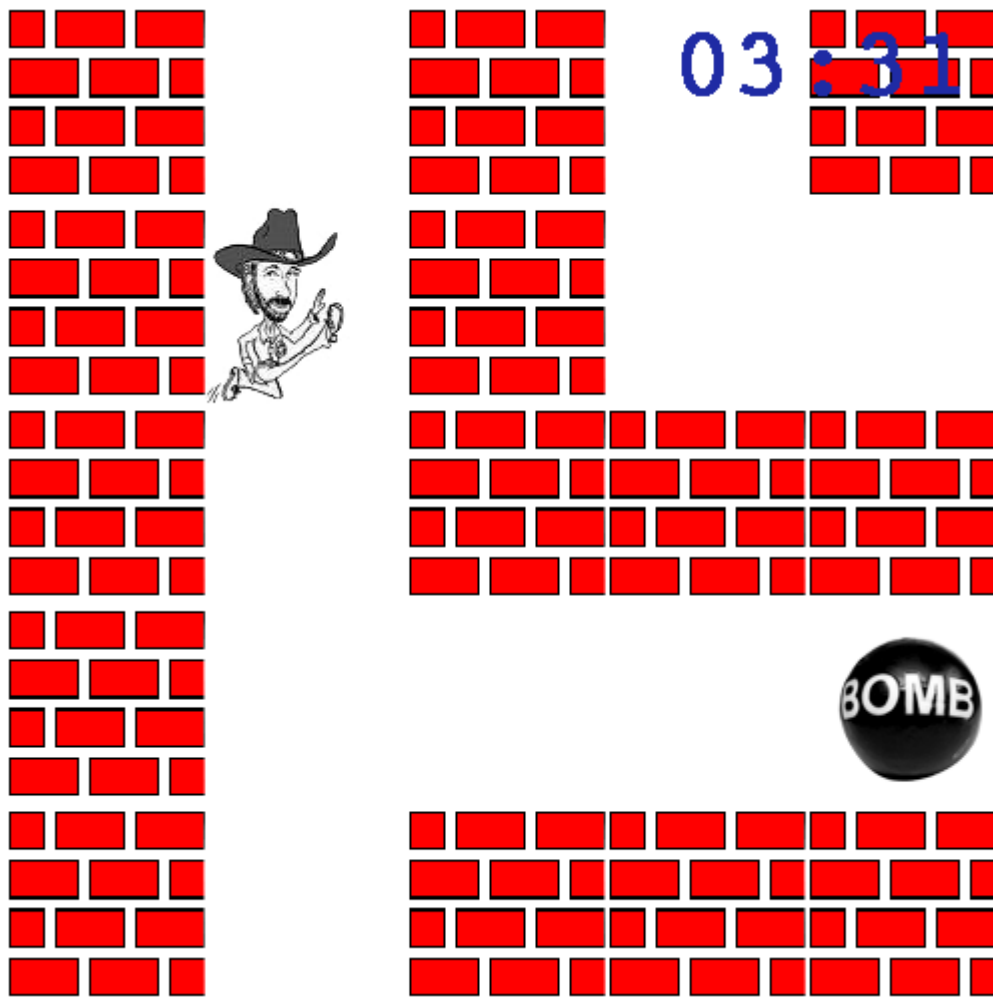
- „Singleplayer“:
Startet den „Singleplayer-Modus“ (siehe 7.2)
- „Multiplayer“:
Startet den „Multiplayer-Modus“ (**OPTIONAL** – Möglichkeit für eine eventuell später folgende Erweiterung)
- „Options“:
Ruft das Optionsmenü auf (siehe 7.4)
- „Quit“:
Beendet die Anwendung bzw. das Spiel

7.2 Singleplayer



- „Character“:
Charakterauswahl (es stehen die in Punkt 4 beschriebenen Charaktere zur Auswahl)
- „Size of Field“:
Spielfeldgröße (Wie groß soll das Spielfeld in Bezug auf die Realität sein?)
- „Difficulty“:
Schwierigkeitsgrad (beeinflusst die Anzahl an Verzweigungen bzw. Sackgassen sowie Gegenständen im Labyrinth)
- „Generate Labyrinth and start Playing“:
Der Labyrinth-Generieralgorithmus (siehe Punkt 4) wird ausgeführt und das Spiel anschließend gestartet.

7.3 Spieloberfläche



- Der Spieler versucht sich den Weg durchs Labyrinth zu bahnen (in diesem Fall mit dem Charakter „Nuck Chorris“).
- Am Weg liegen Gegenstände (in diesem Fall eine „Bomb“).
- Oben rechts läuft die Zeit, die noch verbleibt bis zum „Gameover“, also bis zur Beendigung des Spiels.

8 Qualitätszielbestimmungen

	sehr wichtig	wichtig	weniger wichtig	unwichtig
Sicherheit	X			
Zuverlässigkeit		X		
Korrektheit		X		
Benutzerfreundlichkeit	X			
Effizienz		X		
Portierbarkeit	X			

9 Globale Testszenarien und Testfälle

Testfälle:

- Spieler befindet sich auf dem Weg
- Spieler befindet sich in einer Mauer
- Spieler befindet sich auf einem Gültigen Checkpoint
- Spieler befindet sich auf einem Ungültigen Checkpoint
- Spieler befindet sich auf dem Ziel
- Highscore wird an den Server gesendet

10 Entwicklungsumgebung

Die von uns verwendeten Entwicklungsumgebungen sind:

- Eclipse (Java)
- Visual Studio (.Net)
- Oracle Worksheet

11 Ergänzungen